

EE / CprE / SE 492 - sddec22-13

Simultaneous Call Transmission

Bi-Weekly Report 1

9/02/2022 - 9/16/2022

Client: Collins Aerospace

Faculty Advisor: Dr. Andrew Bolstad

Team Members:

Sullivan Jahnke - Project Manager and Machine Learning Co-Lead

Json Rangel - Reports, Webmaster, and Communication Systems Co-Lead

Tyler Mork - Reports and Communication Systems Co-Lead

Austin Rognes - Research and Machine Learning Co-Lead

Hani El-Zein - Digital Signal Processing Lead and Research

Past Week Accomplishments:

- Sully: Code Modularization and TensorBoard Integration
 - Changed the layout of our master branch by separating each piece into a well-named directory. The main python function lives in the root directory. Since we are referencing a couple different files in our algorithm (training data, modules, models, etc.), this is important to keep it consistent.
 - Created a python class for the neural network implementation and a DataShaper. This allowed for more configurable variables that we can play around with such as `window_jump` (how far our sliding window jumps per sample), `input_size` (the size of the neural network's input layer), `batch_size` (number of samples per training iteration), and `epochs` (number of times the model trains on a batch). To change these variables, just modify the variables under the `'#### Constants ####'` comment in `main.py`.
 - This is also useful for development in that we can just modify a class and it doesn't break the main function.
 - Integrated TensorBoard (<https://www.tensorflow.org/tensorboard>) into our project. This tool allows us to configure metrics in Keras that will then be logged and viewable in a locally hosted web application. Previously, we would just graph metrics using `matplotlib` and as soon as you exited the graph, it was gone forever. There are many good uses for this. First and foremost it allows us to see

how well our models are training/performing with whatever metrics we want to specify. Since logs are saved, we can now compare metrics from different training iterations to see if improvements were made.

- Json: Simulated Data Labeling & Simulink File Updates
 - Worked on determining ways to label generated simulated time-delay signal data.
 - Explored many options, such as adding an array of ones and zeros separate from timeseries data in .mat file.
 - Successfully extracted timeseries data from .mat files.
 - Explored and analyzed complex timeseries data.
 - Successfully extracted complex data into a .txt file.
 - Pushed missing Q-phase test data to personal Git branch.
 - Submitted merge request to master.
 - Updated Simulink simulations to newest version of MATLAB (2022a).
- Tyler:
 - Configured Simulink model to allow for time delayed signal interference. Primary signal is playing the entirety of the simulation with an interferer becoming present at the halfway point. A .mat data file was produced and pushed to Software.
 - Implemented an AWGN channel to sum noise to the output signal of the simulation. Currently, the noise is a White Gaussian noise reduced by 500x in strength.
- Austin: Continuous Integration Pipeline
 - Set up a GitLab .yaml file. Whenever a commit is pushed to master, ssh into the linux-5.ece Iowa State server, git clone the repo, run matlab to generate training data, and git add/commit/push it into the training data folder. Everything works so far except for running the matlab simulink file, which is almost there, just an issue with file paths.
- Hani:
 - Tinker with simulink file, research on doppler

Pending Issues:

- Since we have to shape our data into samples of the input layer size (64 right now), labels for those data points could vary. We need to define how a sample of data is labeled based on the patterns that are present in the labels associated with that 'window'.
- Data is labeled within the python script. It should be labeled as it is generated within the simulation.
- We do not have a dataset to test the model on. This is important to see how the model is performing and would give us more metrics to visualize in TensorBoard.
- The pipeline cannot run the simulink file from the server yet.

Individual Contributions:

Team Member	Contribution	Hours Spent	Total Hours
Sullivan Jahnke	Code Reorganization, Modularization, and TensorBoard Integration	9	40.5
Json Rangel	Data extraction	4	36
Tyler Mork	Noise Implementation Time-Delay Implementation	6	41.5
Austin Rognes	Continuous Integration Pipeline	6	32
Hani El-Zein	Tinkered with simulink model	2	28

Plans for Coming Week:

- Sully
 - Create a python class that takes a window of data with the associated labels, and outputs a single label for that window based on some sort of parameter(s).
 - Integrate the newly formatted data into the algorithm.
 - Play around with the validation split (how much of the training data is used for validation) to see if it affects the model's performance.
 - Create a method in the Neural Network class to use for testing the model. This should just call a keras function with a few added parameters to define the dataset and anything else that function may need..
- Tyler
 - Create standardized SNR for simulation purposes with proposed tolerances for variation. Implement a new White Gaussian Noise in relation to calculated SNR.
 - Provide mechanisms within Simulink to properly label data with an interferer present. Goal is to provide another vector column to the software team that denotes a logical 1 or 0 for interferer or non-interferer data respectively. This is primarily for training purposes. Json will attempt to duplicate the result within Matlab instead of Simulink.
 - Begin implementation of Doppler Effect once the Software team has usable data.
- Json
 - Continue to extract and analyze simulated data.
 - Label data once extracted.
 - Export to suitable file format for Software team (.csv).

- Push labeled data to Git for Software to test.
 - Generate more data.
- Austin
 - Figure out issue with Matlab Simulink paths when running .slx files.
 - Find other uses for pipeline such as testing and verification
 - Make input .wav sound files modular/interchangeable
- Hani
 - Research and attempt to incorporate doppler effect to the team's model